Listing of claims:

1. (Original) A method for performing a product configuration, the product configuration associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

receiving user input specifying at least one selected domain member;

propagating the constraints over the received user input thereby producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member; and

modifying the result by detecting and eliminating incompatibilities caused solely by bounceback behavior.

2. (Original) The method of claim 1, further comprising:

generating a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page; and

providing the configuration page to the user.

3. (Original) The method of claim 1, further comprising:

repeating steps included in the method until the product configuration is complete.

4. (Previously Presented) The method of claim 1, wherein the method is implemented by a set of software instructions running on a computer.

5. (Original) A system for performing a product configuration, the product configuration

associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the system comprising:

a configuration engine adapted to receive user input specifying at least one selected domain member and to propagate the constraints over the received user input thereby producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member; and

a bounceback detection module operatively coupled to the configuration engine, the bounceback detection module adapted to modify the result by detecting and eliminating incompatibilities caused solely by bounceback behavior.

6. (Original) The system of claim 5, further comprising:

a page generation module operatively coupled to the configuration engine, the page generation module adapted to generate a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page, and to provide the configuration page to the user.

7. (Original) The system of claim 5, wherein the configuration engine and the bounceback detection module are implemented by a set of software instructions running on a computer.

8. (Original) A method for performing a product configuration, the product configuration associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

receiving user input specifying at least one selected domain member;

propagating the constraints over the received user input thereby producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member;

modifying the result by detecting and eliminating incompatibilities caused solely by bounceback behavior;

generating a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page;

B2

providing the configuration page to the user; and

repeating the receiving, propagating, modifying, generating, and providing steps until the product configuration is complete.

9. (Original) A method for detecting bounceback behavior associated with a configuration problem, the configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

receiving a domain member selection for a particular variable;

setting a bounceback detection bit vector associated with each non-selected domain member of the particular variable so that each of those bounceback detection bit vectors indicates bounceback behavior;

setting an elimination flag associated with each non-selected domain member of the particular variable so that each of those elimination flags indicates that its associated domain member is tentatively eliminated;

propagating the constraints to identify eliminated domain members of the variables;

setting the bounceback detection bit vector of the eliminated domain members to indicate which variable caused their elimination; and

setting the elimination flag of each of the other eliminated domain members.

10. (Original) The method of claim 9, further comprising preliminary steps of:

B

initializing the bounceback detection bit vector for each domain member of each variable; and

initializing the elimination flag for each domain member of each variable.

11. (Original) The method of claim 9, wherein the receiving step includes receiving a plurality of domain member selections associated with a corresponding number of particular variables, and the setting and propagation steps of the method are performed for each of the domain member selections.

12. (Original) The method of claim 9, wherein bounceback detection bit vectors that indicate bounceback behavior indicate that the particular variable associated with the selected domain member is responsible for elimination of the non-selected domain members.

13. (Original) The method of claim 9, further comprising:

confirming the tentative elimination of a non-selected domain member in response to the bounceback detection bit vector associated with that non-selected domain member not indicating bounceback behavior as a result of subsequent constraint propagation.

14. (Original) The method of claim 9, further comprising:

overriding the tentative elimination of a non-selected domain member in response to the bounceback detection bit vector associated with that non-selected domain member indicating bounceback behavior despite subsequent constraint propagation.

B2

15. (Original) The method of claim 9, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes:

based on the constraints, identifying a domain member causing the eliminated domain member to be eliminated; and

copying the bounceback detection bit vector associated with the identified domain member to the bounceback detection bit vector associated with the eliminated domain member.

16. (Original) The method of claim 9, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes:

based on the constraints, identifying a join corresponding to a disjunction;
logically ANDing the bounceback detection bit vectors associated with the domain members included in the join thereby producing a resulting bounceback detection bit vector; and
copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member.

17. (Original) The method of claim 9, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes:
based on the constraints, identifying a join corresponding to a conjunction;
logically ORing the bounceback detection bit vectors associated with the domain members included in the join thereby producing a resulting bounceback detection bit vector; and
copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member.

B2

18. (Original) The method of claim 9, further comprising:
generating a configuration page based on the constraints so that domain members identified as being eliminated due to bounceback behavior are not marked as conflicted choices on the configuration page; and
providing the configuration page to a user.

19. (Original) The method of claim 9, wherein the steps of the method are repeated each time a user submits one or more new domain member selections.

20. (Previously Presented) The method of claim 9, wherein the method is implemented by a set of software instructions running on a computer.

21. (Original) A method for detecting and eliminating bounceback behavior associated with a configuration problem, the configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable, the method comprising:

initializing a bounceback detection bit vector for each domain member of each variable;

initializing a elimination flag for each domain member of each variable;

B2

receiving a domain member selection for a particular variable;

setting the bounceback detection bit vector associated with each non-selected domain member of the particular variable so that each of those bounceback detection bit vectors indicates bounceback behavior;

setting the elimination flag associated with each non-selected domain member of the particular variable so that each of those elimination flags indicates that its associated domain member is tentatively eliminated;

propagating the constraints to identify eliminated domain members of the variables;

setting the bounceback detection bit vector of the eliminated domain members to indicate which variable caused their elimination; and

setting the elimination flag of each of the other eliminated domain members.

22. (Original) The method of claim 21, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes:

based on the constraints, identifying a domain member causing the eliminated domain member to be eliminated; and

copying the bounceback detection bit vector associated with the identified domain member to the bounceback detection bit vector associated with the eliminated domain member.

B2

23. (Original) The method of claim 21, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes:

based on the constraints, identifying a join corresponding to a disjunction;

logically ANDing the bounceback detection bit vectors associated with the domain members included in the join thereby producing a resulting bounceback detection bit vector; and

copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member.

24. (Original) The method of claim 21, wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused  that domain member's elimination includes:

based on the constraints, identifying a join corresponding to a conjunction;

logically ORing the bounceback detection bit vectors associated with the domain members included in the join thereby producing a resulting bounceback detection bit vector; and

copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member.

25. (Currently Amended) The method of claim 21, wherein the method is implemented by a set of software instructions running on a computer.

B2

26. (New) A method for performing a product configuration, the method comprising:

receiving user input specifying at least a first domain member of a plurality of domain members, the plurality of domain members being associated with a variable of the product configuration;

propagating one or more constraints associated with the received user input to produce a result that identifies a potential incompatibility of a second domain member of the plurality of domain members, the one or more constraints characterizing limits on the product configuration; and

modifying the result by eliminating the potential incompatibility if the potential incompatibility is caused solely by the specification of the first domain member and constraint propagation resulting from the specification of the first domain member.

27. (New) The method of claim 26, wherein the constraint propagation involves at least one other variable of the product configuration.

28. (New) A system for performing a product configuration, the system comprising:

means for receiving user input specifying at least a first domain member of a plurality of domain members, the plurality of domain members being associated with a first variable of the product configuration;

means for identifying an incompatibility in the product configuration, the incompatibility being based on the user input and propagation of constraints of the product configuration, the identification excluding any incompatibilities caused solely by bounceback behavior; and

producing a result including the identified incompatibility.

29. (New) A computer readable medium including computer code stored thereupon, the computer code comprising:

B2

a code segment configured for receiving user input specifying at least a first domain member of a plurality of domain members, the plurality of domain members being associated with a variable of the product configuration;

a code segment configured for propagating one or more constraints associated with the received user input to produce a result that identifies an potential incompatibility of a second domain member of the plurality of domain members, the constraints characterizing limits on the product configuration; and

a code segment configured for modifying the result by eliminating the potential incompatibility, if the potential incompatibility is caused solely by the specification of the first domain member and constraint propagation resulting from the specification of the first domain member.

13 2